**AI-ML Assignments**

**step1: Importing all Packages**

we need to import below packages before we start analyzing any data

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

data1 = pd.read\_excel("general\_data.xls",sheet\_name=0)

data1.head() # this function displays first 5 Rows

Out[5]:

Age Attrition ... YearsSinceLastPromotion YearsWithCurrManager

0 51 No ... 0 0

1 31 Yes ... 1 4

2 32 No ... 0 3

3 38 No ... 7 5

4 32 No ... 0 4

[5 rows x 24 columns]

data1.columns #display all columns present in sheet

Out[6]:

Index(['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager'],

dtype='object')

**Step 2 : Data Treatment**

In Data Treatment we remove null and Duplicate Records from Excel file

data1.isnull() # To check Null values is present in excel sheet or not

Out[7]:

Age Attrition ... YearsSinceLastPromotion YearsWithCurrManager

0 False False ... False False

1 False False ... False False

2 False False ... False False

3 False False ... False False

4 False False ... False False

... ... ... ... ...

4405 False False ... False False

4406 False False ... False False

4407 False False ... False False

4408 False False ... False False

4409 False False ... False False

[4410 rows x 24 columns]

data1.dropna() This function is used to Remove Null Values from Excel sheet

Out[8]:

Age Attrition ... YearsSinceLastPromotion YearsWithCurrManager

0 51 No ... 0 0

1 31 Yes ... 1 4

2 32 No ... 0 3

3 38 No ... 7 5

4 32 No ... 0 4

... ... ... ... ...

4404 29 No ... 1 5

4405 42 No ... 0 2

4406 29 No ... 0 2

4407 25 No ... 1 2

4408 42 No ... 7 8

[4382 rows x 24 columns]

data1.duplicated() This function is used to Check duplicate value present in excel

Out[9]:

0 False

1 False

2 False

3 False

4 False

4405 False

4406 False

4407 False

4408 False

4409 False

Length: 4410, dtype: bool

data1.drop\_duplicates() This function is used to drop duplicates values from data

Out[10]:

Age Attrition ... YearsSinceLastPromotion YearsWithCurrManager

0 51 No ... 0 0

1 31 Yes ... 1 4

2 32 No ... 0 3

3 38 No ... 7 5

4 32 No ... 0 4

... ... ... ... ...

4405 42 No ... 0 2

4406 29 No ... 0 2

4407 25 No ... 1 2

4408 42 No ... 7 8

4409 40 No ... 3 9

[4410 rows x 24 columns]

**Step 3 : Univariate analysis**

Describe Function ()-

descdata=data1[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].describe()

output :



Mean function :

meandata1=data1[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

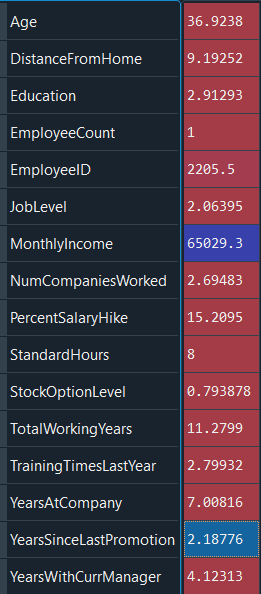
'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].mean()



median function:

mediandata=data1[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

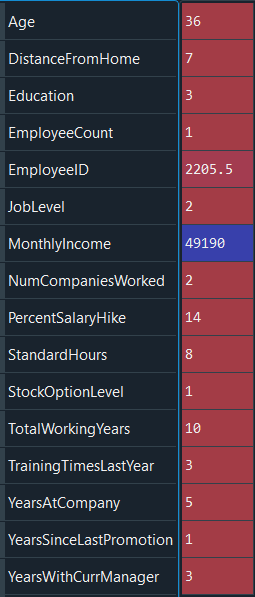
'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].median()



Mode Function () :

modedata=data1[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].mode()

|  |  |
| --- | --- |
| Age | 35 |
| Attrition | No |
| BusinessTravel | Travel\_Rarely |
| Department | Research & Development |
| DistanceFromHome | 2 |
| Education | 3 |
| EducationField | Life Sciences |
| EmployeeCount | 1 |
| EmployeeID | 1 |
| Gender | Male |
| JobLevel | 1 |
| JobRole | Sales Executive |
| MaritalStatus | Married |
| MonthlyIncome | 23420 |
| NumCompaniesWorked | 1 |
| Over18 | Y |
| PercentSalaryHike | 11 |
| StandardHours | 8 |
| StockOptionLevel | 0 |
| TotalWorkingYears | 10 |
| TrainingTimesLastYear | 2 |
| YearsAtCompany | 5 |
| YearsSinceLastPromotion | 0 |
| YearsWithCurrManager | 2 |

std():

stddata=data1[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

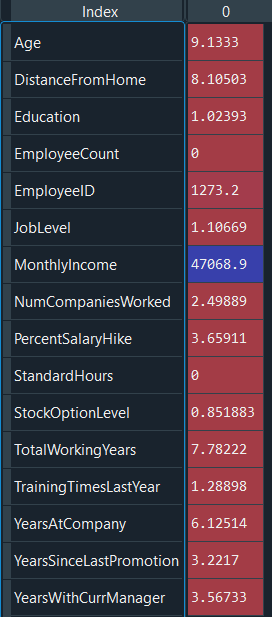
'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].std()



Var():

vardata=data1[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

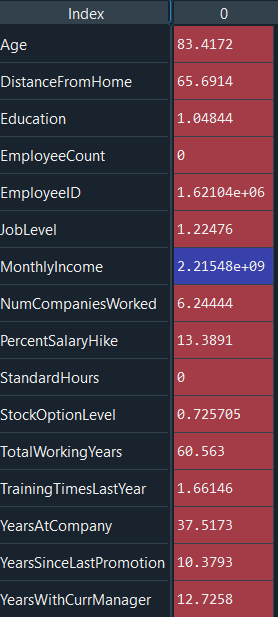
'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].var()



Skew Function :

skew=data1[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

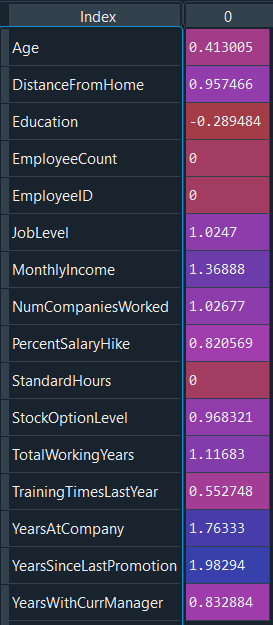
'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].skew()



Kurt():

kurtdata=data1[['Age', 'Attrition', 'BusinessTravel', 'Department', 'DistanceFromHome',

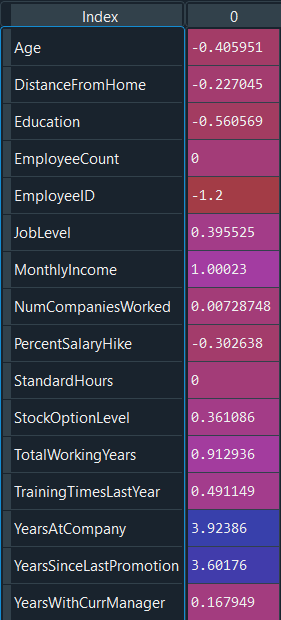
'Education', 'EducationField', 'EmployeeCount', 'EmployeeID', 'Gender',

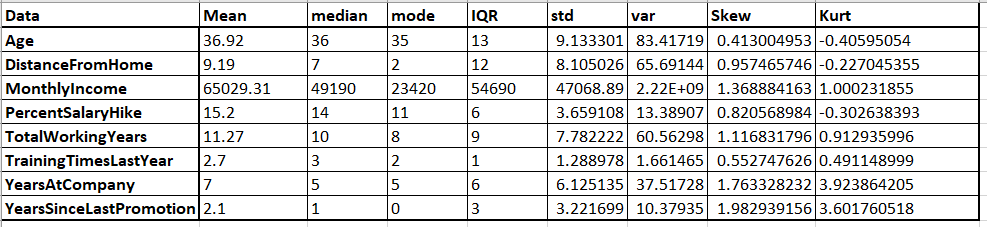
'JobLevel', 'JobRole', 'MaritalStatus', 'MonthlyIncome',

'NumCompaniesWorked', 'Over18', 'PercentSalaryHike', 'StandardHours',

'StockOptionLevel', 'TotalWorkingYears', 'TrainingTimesLastYear',

'YearsAtCompany', 'YearsSinceLastPromotion', 'YearsWithCurrManager']].kurt()





Inference from analysis:

we can see that all variables are Positive skewed

Precent salary Hike ,Age and Distance from is leptokurtic and all others variables are platykurtic

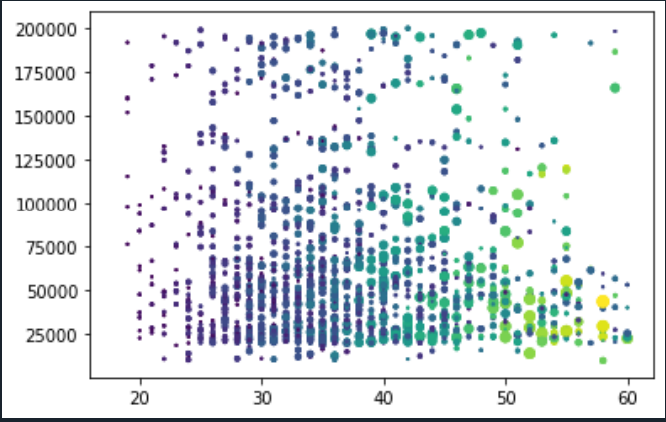
**Outliers**

ScatterPlot :

plt.scatter(data1.Age,data1.MonthlyIncome,data1.YearsAtCompany,data1.TotalWorkingYears)

There’s no regression found while plotting Age, MonthlyIncome, TotalWorkingYears,

YearsAtCompany, etc., on a scatter plot

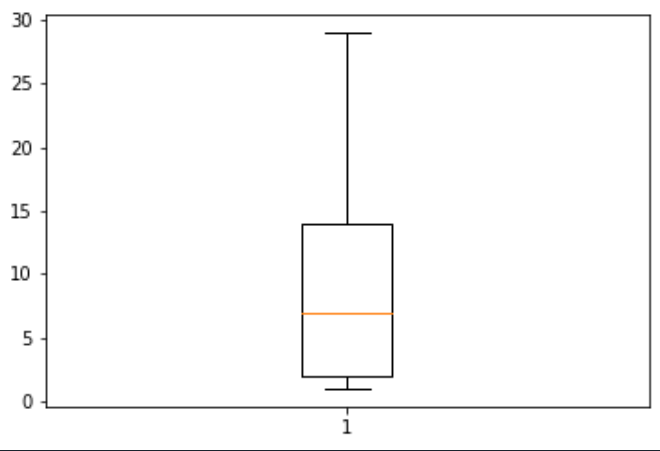


Boxplot:

box\_plot=data1.DistanceFromHome

DistanceFromHome

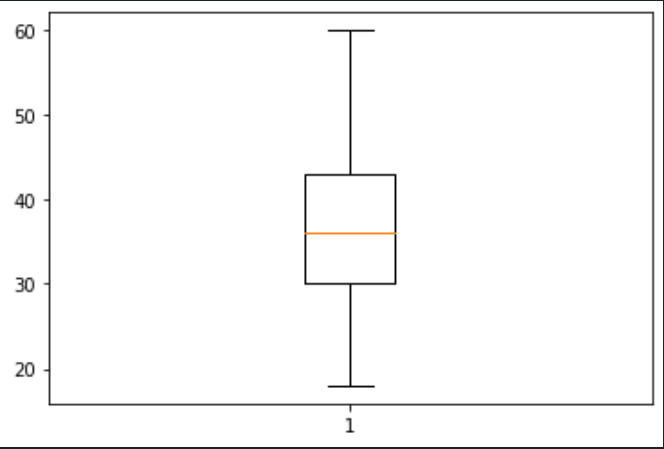
plt.boxplot(box\_plot)



Distance from Home is equally distributed without any outliers.

Age

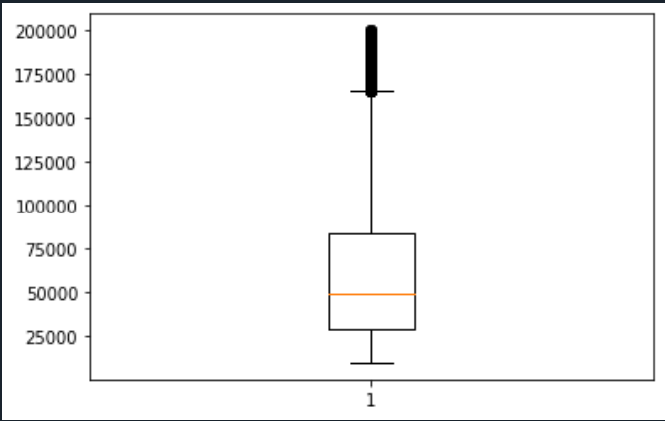
plt.boxplot(data1.Age)



Age is equally distributed without any outliers.

Monthly Income

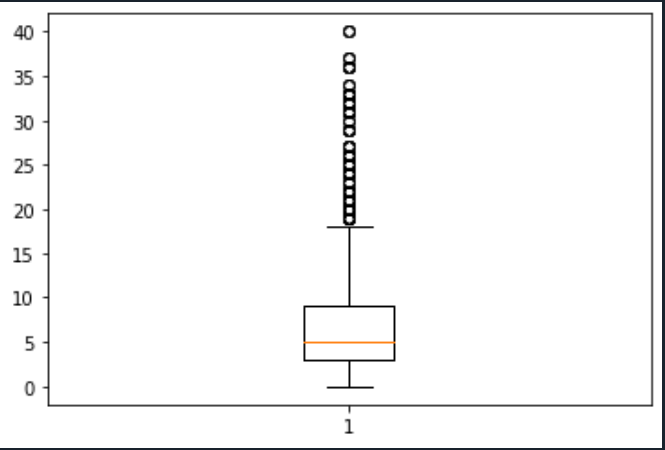
plt.boxplot(data1.MonthlyIncome)



Monthly Income is Right skewed with several Outliers.

YearsAtCompany :

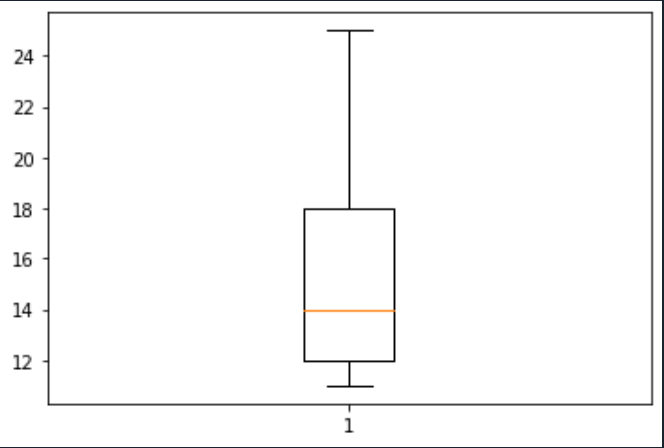
plt.boxplot(data1.YearsAtCompany)



YearsAtCompany is right skewed with many outliers.

PercentSalaryHike:

plt.boxplot(data1.PercentSalaryHike)



Percent Salary hike is equally Distributed without any Outliers